A SIMPLIFIED APPROACH TO CHARACTER RECOGNITION

Patrick Leo Townsend

United States
Naval Postgraduate School

THESIS

A Simplified Approach to Character Recognition

by

Patrick Leo Townsend

Thesis Advisor: G. D. Gibbons

June 1971

A Simplified Approach to Character Recognition

by

Patrick Leo Townsend
Captain, United States Marine Corps
B.S., Marquette University, 1963

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1961

ABSTRACT

This thesis describes an off-line character recognition procedure, tested on the ten digits. The basic scheme is an attempt to recognize the character on the basis of a general memory scheme dependent on the overall shape of the entire character. Failing this, a system of checking two specific features, loops and spurs, is called into play on an as-needed basis. This idea of considering specific features only if a more general recognition procedure does not result in a single answer is unique in the character recognition field.

Comparison to other systems indicates that the system is relatively successful, particularly in view of the reduced computer effort expended.

TABLE OF CONTENTS

I. INTRODUCTION

Since the mid-1950's, computerized character recognition has been a matter of interest to both the business and academic communities. The business applications initially required very highly controlled input, such as the E13B magnetic font on checks, but have been progressively widening their range of allowable inputs. The Postal Office's Zip Code Reader is a good example of a recent successful Optical Character Reader [1]. The academic approaches have been more theoretical in nature and have concentrated on hand-printed characters.

This thesis presents a new approach to the recognition of hand-printed numerals. It is argued that previous methods of recognition have been unnecessarily complex, due in part to a dissimilarity to human recognition procedures. It was the author's experience when first beginning to explore this field that these systems were difficult to comprehend, apply, or research. The procedure detailed here is simple in concept, more nearly imitative of the conscious human process[1], and comparatively simple to duplicate. It is felt that potential improvements to the system could raise its accuracy level to the high ninety's. Areas which appear most susceptible to improvement are pointed out below. In its present form, its success rate is 93%. These results were obtained using a set of characters which consisted of fifty examples of each of the ten digits. The five hundred test numerals were obtained from A.L. Knoll of Honeywell

---

[1]The author's concept of the conscious human method of character recognition is explained in section II.

Industries [2], and consist of characters presented on a twenty-one by twenty-five binary matrix. A sample of three successfully recognized characters is shown in figure 1.

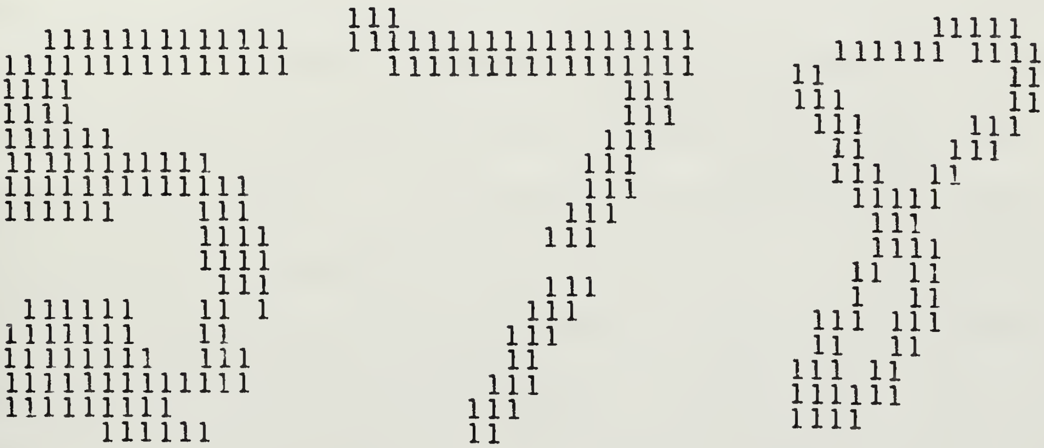


Fig. 1A-"5" Fig. 1B-"7" Fig. 1C-"8"

This thesis is presented in three parts. First, a statement of the basic ideas. Second, five other specific systems are outlined. A detailed description of the present system is given in part three.

II. BASIC IDEAS

The basic idea of the proposed system was gleaned from a conversation between the author and his wife. Asked how she recognized characters, her reply was, "I don't think about it, I know them." Given the hypothetical situation that a character was drawn so as to be confusing, she theorized that each character had specific features which aided recognition. The scheme of this system is: 1) An attempt to recognize the character with a fast and simple procedure, 2) In case of confusion, i.e., an inability to distinguish between two or more possible values for a character, a determination based on specific features. It remained to find a procedure that would accomplish step one, as the typical system uses only step two, i.e., a list of specific features is obtained prior to attempting to recognize the character.

A survey of material on the methods of teaching character recognition to children [3, 4, and 5] also indicated that this idea is a possible description of the human process of character recognition. In every case, learning of the characters is taught through repetition and memorization of the overall appearance of the character as it appears within a word. The most specific instruction suggestion is in [2], when, in the case of frequently confused letters, memory devices such as the following are offered:

> "This is b
> b is on the line
> b is tall like a building
> b looks to the right"

In other words, current teaching practice is to have children memorize the characters' appearance and, in case of confusion, look for specific

appropriate features. The latter is of particular importance when asking the children to reproduce characters in their own hand [6].

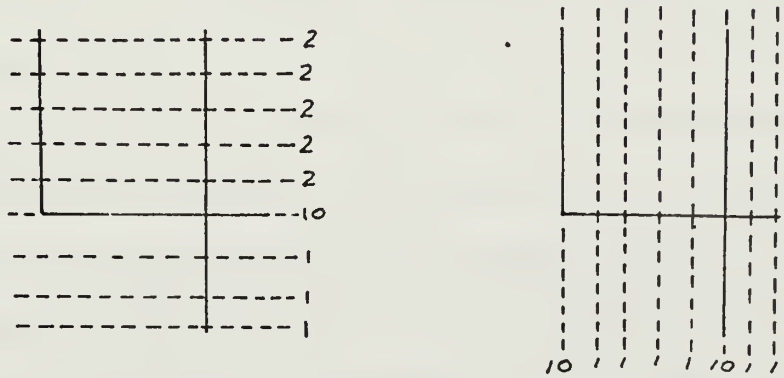
The proposed system immediately knows most numerals based on a simple memory scheme involving two very general parameters. These two parameters were originally suggested by the method described in [7] and below. They are the row and column vectors and are dependent upon the overall configuration and structure of a character. These parameters are easily determined and correct recognition was possible, using only these parameters, in 78.8% of the cases. The decision to rely heavily on this limited amount of information was based initially on a hand-simulation of the parameters on a set of carefully drawn characters. It was observed that even when the drawn characters were allowed to deviate from the norm, at least one of the parameters was almost invariably equal, or nearly equal, the ideal. From this it was deduced that when use of the parameters could not determine a single answer, it would enable the compilation of a narrowed list of possibilities.

The second step is checking for specific features, namely closed loops and spurs. Only those features which will help to differentiate between particular choices will be considered. If for instance, the row and column vectors narrowed the possibilities to "0" and "9," it would be superfluous to determine if the character has any closed loops. Nor would a human, once he had decided that a character was either a "0" or a "9," look at the loop. In this case, the feature which facilitates recognition for both the human and the machine is the presence of a spur.

Using this information as a basis, the proposed system consists of: 1) a general memory scheme that will consider the character as a whole, and 2) a dynamic system of checking specific features. Throughout this

paper, "dynamic" is used to mean "on an as-needed basis," i.e., certain steps will be performed only when a need to perform them has been determined. As previously noted, only step one was necessary in almost 80% of the cases.

The consideration of the character as a whole consists of intersecting the character with a series of horizontal and vertical lines and keeping track of how many lines each of the rays intersects. Figure 2 illustrates this process.



2A. Horizontal Scan Vector 2B. Vertical Scan Vector

The resulting vectors are then compressed by eliminating successive duplications and all zeroes. In this way, the horizontal scan vector of figure 2A becomes 2-10-1, while the vertical vector is 10-1-10-1. These two vectors can now be compared with the vector pairs in the memory and the answer "4" obtained.

In some cases the vectors obtained from a character are not adequate matches with any of the standard sets. No answer results, but the choice is narrowed, usually to two or three possibilities, on the basis of closeness of fit. The closeness of fit is determined by a least squares system which is explained in Section IV of this paper.

III. CURRENT SYSTEMS

Among five current systems chosen for comparison in this thesis, none use the basic approach just described. The five were chosen to be included because of their wide divergence and relative success.

The procedure presently in use by the Japanese Postal Department is based on a horizontal scan technique far more complicated than simply counting the lines crossed by each matrix row [7]. A three by three window moves to the right one column at a time, highlighting each successive portion of the character. The pattern within the window is categorized, the information stored, and the window shifted. At the completion of each horizontal scan the window is moved back to the left edge, down two rows, and again begins shifting to the right. The extracted features are identified as belonging to a particular character by utilizing a set of sequential decision diagrams. Due to the divergent ways of legitimately drawing any given number, several transition diagrams are used for each number.

The authors do not give any statistical results, but only state that:

> "...Three models have been installed in Tokyo and Osaka Central Post Offices and constantly field tested to achieve high performance levels..."

It can be reasonably assumed that their recognition percentage must be at least in the mid-nineties.

Munson's system for character recognition was developed and tested on a forty-six character alphabet, rather than the ten character set of digits used by the other systems [8]. Because of the increased possible choices and the number of characters which are difficult even for a human

to differentiate between out of context, e.g., "C" and "(," the longer the alphabet, the more difficult the problem. In view of this, the fact that Munson's procedure is the most complicated is not surprising. If the proposed system were to attempt the same forty-six character alphabet, the percentage of successful recognition would surely drop, but not, it is felt, significantly below Munson's. The idea of considering context when recognizing a character, a procedure which greatly eases the human recognition problem, is not utilized by any of the systems discussed in this thesis, but is detailed in [8].

Two exhaustive procedures are used to extract feature characteristics of the character under consideration. The first of these, PREP, is "... a simulation of a previously constructed optical preprocessor capable of extracting, in parallel, 1024 optical correlations between a character image and a set of photographic templates, or masks." The second, TOPO, "... extracted a large number of topological and geometric features of the character image."

PREP's output is nine 84-bit feature vectors. Each feature vector describes the location and orientation of edges in each of 84 regions. The nine vectors are the result of presenting each image to PREP nine different times, "... first in the center of the 24X24 field, then in the eight positions formed by translating it vertically and/or horizontally by two bits."

TOPO's output is 68 features, "... 16 for the spurs, 16 for the concavities, 8 for the enclosures, 6 for overall character size and shape, and 22 resulting from special calculations about the width of the character at various levels, discontinuities in the profiles, etc." The first step in TOPO, the construction of a perimeter around, and adjacent to, the character is used in the system which is the subject of this thesis.

Utilizing the information produced by TOPO and PREP to recognize a character is the job of a piecewise linear type learning machine, of the type described by Nilsson.

Munson's results, shown in Appendix A, indicate that both TOPO and PREP are necessary to obtain the very good recognition percentage of 85%. This percentage is particularly noteworthy in view of the length of the alphabet with which it was obtained.

The system using "characteristic loci," devised by Glucksman and improved by Knoll bears a greater resemblance to the present system, although the basic approach is still quite different. The characteristic loci contain five numbers and are formed in the following manner. Starting with a given point in the binary array, the first number of the locus is set equal to the value of the point. The point will, of course, be a one if it is a part of the character and a zero if it is not. Rays are then assumed emanating out from the point to the left, right, up and down. The second value of the locus is set equal to the number of lines crossed by the ray, extended to the left. The third value represents the number of lines crossed by the ray extending straight up, and so on in a clockwise direction. The maximum number of lines crossed is set at two. Knoll found that sixteen or less characteristic loci were necessary to define a digit when the alphabet consisted of only the ten numerals.

The "characteristic loci" features are utilized in two separate recognition schemes. One is an "exact match" while the other is a linear discriminate function scheme. The linear discriminate function scheme is a standard recognition procedure which involves the scalar

product of the characteristic vector determined for the character being tested and the standard vector. Knoll's results with the Honeywell numeric data set are the best of any system described in this paper. Working with Munson's SRI data, Knoll was able to obtain nearly equal results to Munson, when working with just the numerical characters or just the alphabetic characters. This was prior to Munson's adding PREP to TOPO, however. Knoll's results are included in Appendix A. The 98.9% success rate of Knoll's system is significantly better than the 93% achieved by the proposed system in its present form.

The three methods described above all work with characters which have been transferred by hardware from their natural drawn state into a binary matrix. A procedure which works with a character "as is," is described by Greanias, et al [10]. By use of a logic control raster, Greanias and his collegues were able to construct a system which "... recognized 99.3% of numerals written by 45 subjects after thirty minutes of training." There is a degree of dynamic parameter checking in that internal regions are only investigated if a "0," "1," "6," "8," or "9" is indicated as a possibility. The utilization of this additional logic is necessary every time one of these numbers is recognized, rather than on an as-needed basis to clear up confusion.

All of the above methods, and the authors, are off-line, i.e. the drawing of the character and its recognition by the machine are separated by any amount of time that is convenient. One recent on-line attempt at character recognition is that of Powers [11]. As he points out, the main characteristic difference between off-line and on-line recognition, which is the computer knowing the time sequence of the strokes, is a mixed blessing. Knowing in which sequence the lines were drawn would make

differentiating between characters generally easier, except that if a person were to draw a zero clockwise when the machine only expected counter-clockwise zeroes, the machine would fail to recognize the zero, geometrically perfect or not.

Powers worked with the slopes of successive line segments. His task was made more difficult by the fact that he considered only the single parameter of direction sequence. His best percentage of recognition was 92.8% and this was obtained when he was the only subject inputting figures. He admits that

> "... natural ambiguities between the direction sequence descriptions of different characters are resolved by conditioning the user."

His results are included in Appendix A.

All of the systems outlined above share one characteristic. The character is completely analyzed, all parameters are determined, before any final decision making is attempted. In this they are very unlike conscious human behavior.

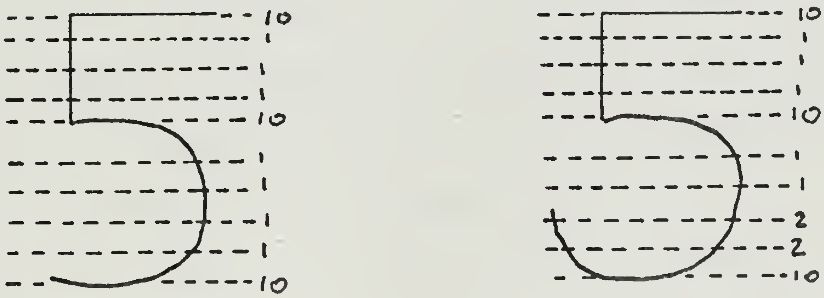
Appendix A shows that the least complicated system, Knoll's, is superior to Powers', and apparently on a par with the far more complex system of Munson. A definite statement cannot be made until all systems have attempted the forty-six character alphabet that Munson's system was applied to. If a dynamic parameter determination system could be incorporated in any of the above systems, the amount of computation, at least, would be reduced.

IV. THE PROPOSED SYSTEM

As with all previous methods discussed, the author's procedure involves (1) establishment of standards, (2) preprocessing of the character being analyzed, and (3) determination of the value, or name, of the character. The proposed standards are few in number and easy to understand.

The five hundred test numerals were drawn from the handwriting of thirteen authors. A copy of the character set will remain with Prof. Gibbons at the United States Navy Postgraduate School.

The primary standards, used in the analysis of every character, are the vertical and horizontal vectors, explained previously. As was seen for the figure "4," the vertical vector was 10-1-10-1, while the horizontal vector was 2-10-1. An area of initial difficulty was determining the vectors for figures with curves. If, for instance, a zero were perfectly round and drawn with a very thin line, the horizontal and vertical vectors would both be 1-2-1. In reality however, curves are always flat enough and lines thick enough so that the vectors are, in fact, 10-2-10 for a zero. Some numbers may require two sets of vectors. This is caused by the fact that in writing numbers with curves, some people follow through more than others. Figure 3 illustrates this possibility.



3A 3B

The difference between the two possible horizontal vectors is very significant.

The first additional standard, considered incases where the primary system fails to determine a single name for the character, is the number of closed loops contained in the figure. This standard is exactly as would be expected. Eight is given a value of two for its two closed loops; six, nine and zero, a value of 1; and all others, a value of 0.

The last standard is the property of having a spur, or loose end, in a particular quadrant. This does not mean that the character must be carefully centered on the matrix but that the character is always considered as if it were split into four quadrants reading counter-clockwise from the upper left. For instance, when scanning from the left to the right and down, if the first line that is encountered is a spur, such as with a "2," then the upper left standard has a value of 1. If, however, the first line encountered is not a spur, such as with a "9," then the upper left standard is 0. Reading counter-clockwise, starting at the upper left, the standards for "3" are 1-1-0-0 while for "0," they are 0-0-0-0.

In the present version of the system, all the above standards are pre-determined and remain static throughout. This is not, of course, mandatory. In an earlier version of this basic idea, using only the vertical and horizontal vectors, implemented on an SDS 9300 with an AGT/1 graphics display, the standards were determined dynamically. With a simple averaging function it could be extended to improve itself and thus become tuned to a particular user's handwriting.

The preprocessing performed on the characters is minimal. It may be considered as compensatory for the inherent problems in reconstructing a

smooth, hand-drawn character on a twenty-five by twenty-one binary matrix. Gaps and irregular lines naturally appear. The scheme of smoothing employed has two steps. First is to widen every line by causing any matrix values immediately to the right and immediately below each "1" in the original matrix to become "1's" also. After this has been done, a second pass through the matrix is made, eliminating obvious irregularities. The procedure is simple. Any time a "1" or a "0" is found bordered on three sides by two or more of the opposite value, the odd one is changed to agree with its surrounding values. Figure 4 illustrates this filling and smoothing process.

11100000	11110000	11110000
11100000	11110000	11110000
10000000	11100000	11110000
11110000	11111000	11110000
11100000	11110000	11110000
11100000	11110000	11110000
4A. Original portion of a line.	4B. After filling.	4C. After filling and smoothing.

This process can be thought of as performing for the machine the compensations that are automatically performed with the human eye when ignoring minor irregularities. A weakness is that the smoothing process applies only to horizontal and vertical lines, not slanted lines.

After the character has been preprocessed, the extended horizontal and vertical vectors of the character are determined. This is accomplished by a straightforward row-by-row and then column-by-column scan of the matrix. Whenever less than seven consecutive ones are encountered, followed by a zero, a counter is incremented. When each row or column is completely scanned, the value of the counter, which represents the number of lines crossed, is stored and the counter is reset to zero. If

seven or more consecutive ones are found, the value of the counter is set at 10 to indicate the presence of a straight line. In this way any line which is longer than 1/3 of the width of the character, or 3/10 of the height is considered a straight line. It was arbitrarily decided not to allow any counter to exceed a value of 10, even if both a straight line and an intersected line were found in a particular row or column. While this caused no apparent discrepancies or problems, it might be an area for investigation.

The next step involves compacting the extended vectors, with one entry for each row or column, into the same form as the standard vectors. The extended vectors are inspected sequentially with the following rules in effect: (1) All zeroes are discarded. (2) Any value except a ten which is unequal to both its immediate predecessor and successor is discarded. (3) Any nondiscarded value which is unequal to the last value inserted into the compacted vector, is added to the compacted vector. For instance, if the beginning of an extended vector were 0-0-1-2-2-2-10-2-2, the rules would be applied in the following manner:

1) The first two zeroes are discarded.
2) The one is discarded since it is unequal to both its predecessor, zero, and its successor, two.
3) The first two remains, causing a value of two to be placed in the compact vector.
4) The next two two's are discarded since the last value added to the compact vector was a two.
5) The ten is added to the compact vector.
6) The next two is added to the compact vector while the one following is discarded.

Figure 5 illustrates this process.

```
 0
 0
 1
 2 ----------------------> 2
 2
 2
10 ----------------------> 10
 2 ----------------------> 2
 2
 .                         .
 .                         .
 .                         .
```

5A. Extended vector   5B. Compact vector

The process of determination can now begin. Two values for each digit are found by comparing the newly established compact vectors with the standard vectors for each of the ten numerals. These values are the row difference and the column difference. The first number in the compact row vector is subtracted from the first number of the standard row vector and the difference is squared. The same is done with the second numbers in each row vector and the squared difference is added to that already determined. This continues through the entire set of vectors. The procedure is then repeated for the column vectors. The idea is of course to have a perfectly formed character that results in a row difference and column difference both equal to zero. Of the five hundred numerals tested, 247 of them were a perfect match.

When the match is less than perfect, two variations are employed. They are "shifting" and choosing a single answer from several lists. These operations, which are explained in detail below, are performed before secondary standards, i.e. spurs and loops, are used. The secondary standards were necessary in only 17.2% of the characters tested.

"Shifting" is performed by discarding the first value in each compact vector and moving all succeeding values forward one place. An example of when this operation is necessary follows. If a person draws a four with the left vertical stroke significantly higher than the right one, the resultant row vector is 1-2-10-1, instead of the standard 2-10-1. The row difference would be computed as $(1-2)^2 + (2-10)^2 + (10-1)^2 + (1-0)^2$ or 147, definitely not a match. Yet there is no confusion to the human eye when a character is drawn in this manner. The only requirement is that the character look more like one numeral than any other. To aid the computer in making this kind of distinction, shifting is introduced. After shifting, the compact row vector of 1-2-10-1 becomes 2-10-1, yielding a row difference of 0.

Before and after shifting, several lists are maintained while sequencing through the standard vectors. Use of these lists is the second variation necessary when the match is not perfect. Comparisons of row and column differences determine which group of standard characters will make up any given list, and no one test character will ever use more than one of the lists. The particular lists employed were established through experience. Several additional lists were included in initial attempts. As work progressed, however, it became obvious that a smaller number of lists was required for an even higher degree of accuracy.

When matching a character's compact row and column vectors against the standard vectors, if the row difference is determined to be zero while the column difference is one, the standard character is added to the "Ziplist" as one possibility for later consideration. The exclusion from the Ziplist of characters with a column difference of zero and a row difference of one was based on experience. Some column vectors are standard for more

than one character, e.g. the standard column vector for "9" and "0" is 10-2-10 and for "5," "6," and "8" is 10-3-10. The standard row vectors are, however, unique. Whenever either the row difference or the column difference is zero, the standard character is added to the "Zerolist." This is the second list (Ziplist being the first) checked for a possible answer. When either the row or column difference is between 0 and 10, the character is placed in the "Lowlist." Differences between 10 and 100 qualify the character for the "Goodlist."

After the new row and column differences are determined, various lists are compiled again as appropriate. If either difference is equal to zero, the "Zerolistsh" (the "sh" suffix indicating post-shift) is added to. If either difference is between 0 and 10, "Lowlistsh" increases in length by one. A difference between 10 and 100 places the standard character on the Goodlist once again.

An entirely new list, "Trylist," is also developed. Requirements for the Trylist are that either the sum of the pre-shift row difference and the post-shift column difference, or the sum of the pre-shift column difference and the post-shift row difference, be less than three. The Trylist is sometimes the third list to be checked for a possible answer.

Choosing an answer from the lists is done in the following sequence:

1) If only one character has been placed in the Ziplist, it is the answer.

2) If only one character has been placed in the Zerolist, it is the answer.

3) If more than one character has been placed in the Zerolist, the Trylist is checked.

4) If the Trylist has but one member, it is the answer.

5) If no answer has yet been found, then Ziplist, Zerolist, Zerolistsh, and Trylist are checked successively to determine if any have multiple entries. The first one found with multiple entries is used. The entries will be further evaluated by use of the additional standard values, loops and spurs.

6) Finally, if necessary, Lowlist, Lowlistsh, and Goodlist are checked in a similar fashion to determine a set of possible choices.

Of the 500 numerals tested, 394 required only the use of row and column vectors to determine a single correct answer. In 20 additional cases, a single answer was produced, but was incorrect. Eighty-six required additional standards. Of these, the correct digit was among the choices determined in 79 cases.

In some cases, where the detection of a loop or a spur is necessary, a preparatory procedure is required.[2] A perimeter of "2's" is constructed around the outer edge of the character. An array is simultaneously built containing the matrix coordinates of the two's, in the order in which they are positioned. Figure 6 shows the results of this procedure in a modified example.

6A. Before perimeter	6B. After perimeter	6C. Sequential perimeter coordinates	
00000	02220	2,1	5,3
01110	21112	3,1	4,4
01010	21012	4,2	3,5
00100	02120	5,1	2,5
01000	21200	6,2	1,4
00100	02120	7,3	1,3
00000	00200	6,4	1,2

---

[2]The basic idea for both this procedure and the subsequent system used in ascertaining the presence of spurs are described in [7].

The coordinates are used in the search for possible spurs. The "2's" are necessary for determining the presence of closed loops. This determination is made by scanning successive columns. A sequence of 2-1-0-1-2, disregarding successive duplications, indicates a closed loop. Appendix B contains an example of a character in its original form, after it has been filled and smoothed, and then after having a perimeter of "2's" constructed around it.

The character can now be checked for loops and spurs as found necessary. As explained briefly above, the use of the closed loop property is straightforward. It is not necessary to determine the presence of any closed loops in the test character unless the fact will aid in discriminating between choices. If, for instance, the choices determined by the row and column vectors all have one loop (i.e., "0," "6," "9"), the presence of any loops in the test character is not even investigated. Similarly, if no loops are present in any of the choices, a closed loop check is not performed. Only in situations in which not all the choices have the same number of loops is a closed loop check of value. When the number of closed loops in the test character in question is determined, all choices that do not have the same number of closed loops are discarded. If one choice remains, it is chosen as the answer. If more than one choice remains, the survivors are checked for spurs. In rare cases, due to an earlier malfunction, no choices remain. The list of possibilities is then reinstated and passed on.

A similar decision logic is employed when checking for spurs. Beginning with the upper left corner and proceeding counter-clockwise, the standards of the remaining possibilities are compared. If all remaining choices have, for instance, ones as their upper left standard, i.e., they all

have upper left spurs, then the upper left characteristic of the character being analyzed is not even determined. Only when there is a possible decision is that facet of the character checked. If, for instance, the choice is between "2" and "3," the upper left would not be checked but a discriminating distinction could be made based on the lower left. In this case, "3" has a spur in the lower left, and "2" does not. After each check for a spur, the remaining possibilities are re-evaluated.

As with closed loops, when one choice remains, it is the answer. If more than one choice remains, they are checked for the next possible discriminating spur. If none remain, the list of choices is reinstated and passed on to the next possible spur check.

The idea of reinstating a list compares with a human reaction, "But it must be one of these." If checking for loops and spurs fails to result in a single answer, a nonrecognition message is printed. This has never occurred.

Results of this system of character analysis are tabulated in Appendix A. The table is constructed to indicate major areas of success and weakness. Many errors attributed to initial decisions or to incorrect choices are in fact caused by the filling and smoothing routines. Sometimes important features are obscured or loops are filled in. However, the filling and smoothing procedures aided accurate character recognition far more than it hindered it. This was determined through a number of tests conducted without the filling and smoothing procedures, during which overall accuracy fell below 90%.

A flow diagram of the proposed system is given in Appendix C.

V. CONCLUSION

A study of Appendix A reveals that the proposed system compares favorably with the other systems described, and exceeds some other systems not described. If accuracy obtained is considered in relation to the amount of computer effort expended or to the complexity involved, the results are significant. The fact that the system is imitative of human recognition procedures and determines parameters dynamically makes it unique.

Possibly the most important feature of the author's system is the ease with which it could be taught. Since the parameters are few in number and easy to understand, an instructor could outline with little difficulty this method of approach to a class just beginning to explore artificial intelligence in general or character recognition in particular. Given a system which is relatively uncomplicated to program, yields a high initial success rate and has some obvious areas for improvement, a student's interest could be caught and held.

When the two weak areas, filling and smoothing of the character, and location of the spurs are perfected, it is also possible that this system would have practical application due to the greatly reduced programming, both hard-wired and soft-ware.

In view of the success of the proposed system, it is felt that future attempts at character recognition should thoroughly investigate comparatively uncomplicated schemes with dynamic determination of parameters, to obtain maximum results.

APPENDIX A

System Results

Investigator and Description	Correct Recog-nition (%)	Rejects (%)	Errors (%)
Case B1: Highleyman [11]—Numeric Data			
Training and testing on original 500 samples	98.8	0.2	1.0
Case B2: Training on original 500			
Testing on 120 new samples	61.6	19.2	19.2
Case C1: Duda and Fossum [12]—Numeric Data			
Training on 500 samples	100.0	—	0.0
Testing on same samples			
Case C2: Training on 400 samples	74.0	—	26.0
Testing on remaining 100			
Case D1: Chow [13]—Alphanumeric Data ($\sim$1800 samples)			
Training on 80 percent of samples	58.3	—	41.7
Testing on remaining 20 percent			
Case E1: Bledsoe [14]—Alphanumeric data			
Training on 80 percent of samples	40.0	—	60.0
Testing on remaining 20 percent			
Case F1: Munson, Duda, and Hart [6]—Alphanumeric Data Nearest-Neighbor Rule			
Based on 80 percent of samples	52.5	—	47.5
Testing on remaining 20 percent			
Case F2: Preprocessing and Piecewise Linear Classification			
Training on 80 percent of samples	68.3	—	31.7
Testing on remaining 20 percent			
Case F3: Same as Case F2, except numeric data only			
(500 samples)	88.0	—	12.0
Case G1: Human Recognition [6]—Alphanumeric Data			
360 samples	84.3	—	15.7
Averages over ten people			

1. Previous Results, Obtained from Ref. 2

Description	Correct Recog-nition (%)	Rejects (%)	Errors (%)
Case A1: Numeric Samples Only*			
394 "good" samples	87.6	6.3	6.1
105 "bad" samples	21.9	22.8	55.3
All 499 samples	73.7	9.8	16.5
Case A2: Upper Case Alphabetics Only**			
854 "good" samples	78.8	10.9	10.3
442 "bad" samples	29.0	39.3	31.7
All 1296 samples	61.8	20.6	17.6
Case A3: Combined recognition results on all 1795 samples	65.1	17.6	17.3

* An Ambiguity resolving procedure has been used.
** Correct recognition includes 122 ambiguities that contain the correct symbol class as one choice. Methods for resolving ambiguities have not been included in the simulation.

2. Performance of Knoll's System
With Standard Highleyman Data, Ref. 2

Description	Recognition Scheme	Correct Recognition Iter 1	Correct Recognition Iter 2	Rejects	Errors
Case 1: 500 Samples (Standard) from 9 Authors	EM	475	21	3	1
Case 2: 352 Samples from Same 9 Authors	EM	344	7	1	0
Case 3: 89 Samples from Author 111	EM	84	4	1	0
Case 4: 89 Samples from Author 113	EM	59	24	6	0
Case 5: 89 Samples from Author 112	EM	86	2	0	1
Case 6: 103 Samples from Author 120	EM	103	0	0	0
Case 7: Totals—1222 Samples	EM	1151 94.20%	58 4.74%	11 0.90%	2 0.16%
		$\underbrace{\hspace{3em}}_{98.94\%}$			
Case 8: 500 Samples (Same as Case 1)	LDF Training	All Correct (convergence after 130 cycles)		—	0
Case 9: 530 Samples (Cases 2, 3, and 4)	LDF Testing	522		—	8

3. Knoll's Results Using Standard Honeywell Data, Ref. 2

Condition	Preprocessor	Number of Iterations	Final Classification Scores: Training Patterns	Test Patterns
Single-Coder File				
1	PREP, 1 view	10	99%	88%
2	PREP, 9 views	27	89%*	96%
3	TOPO	10	94%	91%
4	Combined	--	---	97%
Multiple-Coder File				
2	PREP, 9 views	18	65%*	78%
3	TOPO	4	84%	77%
4	Combined	--	---	85%

4. Munson's Results Using Standard SRI 46-Character Data, Ref. 8

inputs	response 1	2	3	4	5	6	7	8	9	0	N_i	A_i	M_i	X_i
1	11										11	11		
2		8									11	8		3
3			10								11	10		1
4				10							11	10		1
5					9						11	9		2
6						11					11	11		
7							10				11	10		1
8								11			11	11		
9									10		10	10		
0										10	10	10		
totals											108	100	0	8
percentages												92.8%		7.2%

Powers' Results from Ref. 11

N_i = Number of Samples of Character Used

A_i = Number Recognized Correctly

M_i = Number Mis-recognized

X_i = Number of Nonrecognition Errors

	Perfect Match		Single Choice from Lists		Choices Contain Correct Choice		Correct Answer Chosen	Total	
	Right	Wrong	Right	Wrong	Yes	No		Right	Wrong
1	48	0	2	0	0	0	0	50	0
2	13	0	30	1	6	0	6	49	1
3	10	0	19	1	20	0	17	46	4
4	21	0	27	0	2	0	1	49	1
5	11	0	11	2	23	3	21	43	7
6	19	0	15	5	11	0	11	45	5
7	37	0	13	0	0	0	0	50	0
8	12	0	21	4	9	4	7	40	10
9	36	0	9	0	5	0	5	50	0
0	40	3	0	4	3	0	3	43	7
Totals	247	3	147	17	79	7	71	465	35
Percent	49.4%		29.4%				14.2%	93%	

6. Results of Proposed System Using Standard Honeywell Data

APPENDIX B

Example Character

```
00000000000000000000000
00000000000000000000000
00000000000000000000000
00000000000000000000000
00000000000000000000000
00000111111111111100000
00011111111111111100000
00011110000000000000000
00011110000000000000000
00011111100000000000000
00011111111111000000000
00011111111111110000000
00011111100001110000000
00000000000001111000000
00000000000001111000000
00000000000000111000000
00001111110001101000000
00011111110001100000000
00011111111001110000000
00011111111111110000000
00011111111100000000000
00000000111111000000000
00000000000000000000000
00000000000000000000000
00000000000000000000000
00000000000000000000000
00000000000000000000000
```

1. Original

```
00000000000000000000000
00000000000000000000000
00000000000000000000000
00000000000000000000000
00000000000000000000000
00000111111111111110000
00011111111111111110000
00011111111111111100000
00011111110000000000000
00011111110000000000000
00011111111111100000000
00011111111111111000000
00011111111111111000000
00011111100001111100000
00000000000001111100000
00000000000001111100000
00001111111001111100000
00011111111001111000000
00011111111111111000000
00011111111111111000000
00011111111111110000000
00011111111111100000000
00000000111111000000000
00000000000000000000000
00000000000000000000000
00000000000000000000000
00000000000000000000000
```

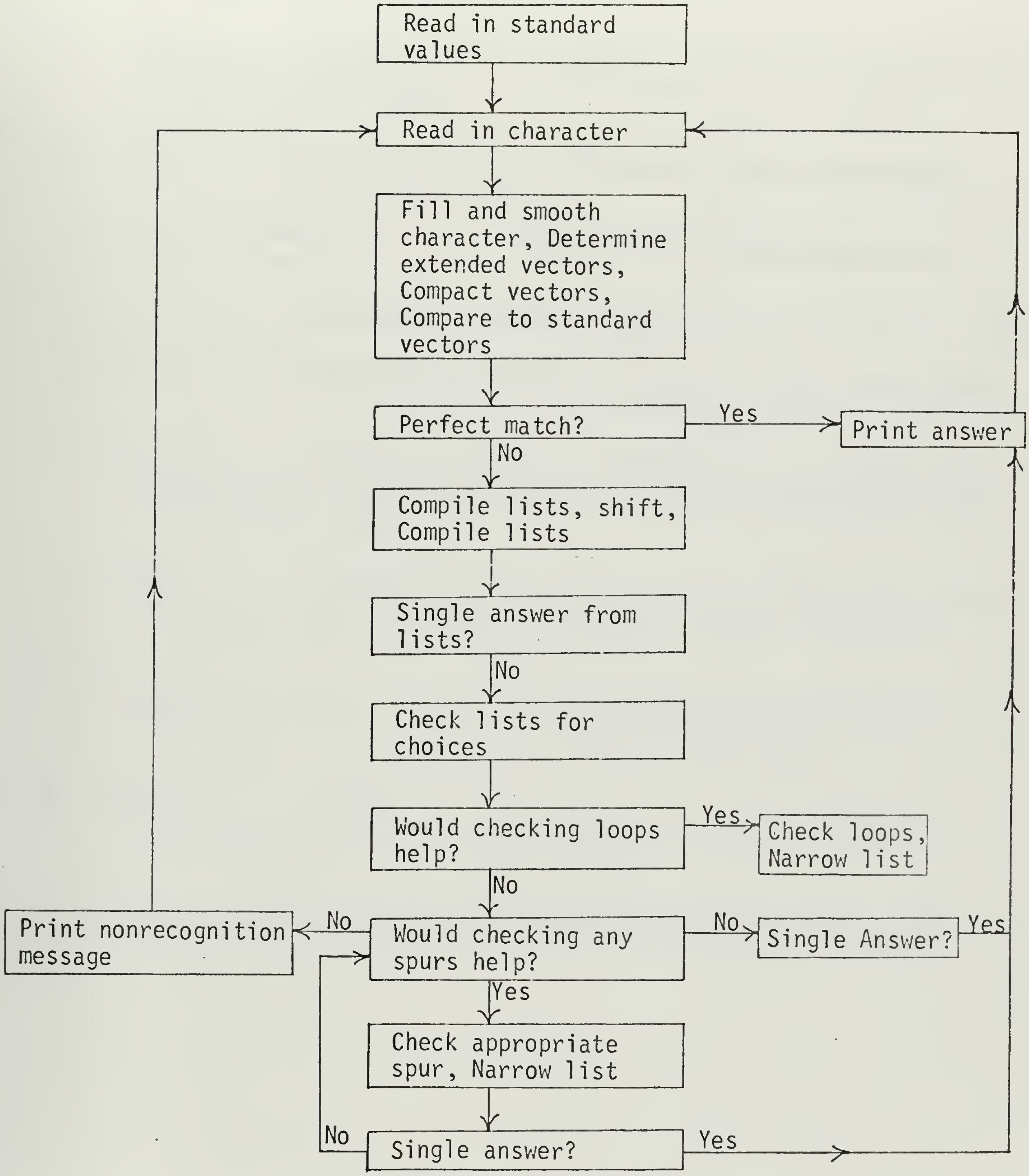
2. After Filling and Smoothing

```
00000000000000000000000
00000000000000000000000
00000000000000000000000
00000000000000000000000
00000222222222222220000
00002111111111111112000
00211111111111111112000
00211111111111111120000
00211111112222222200000
00211111112222200000000
00211111111111122000000
00211111111111111200000
00211111111111111200000
00211111122221111120000
00022222200021111120000
00002222222021111120000
00021111111221111120000
00211111111221111200000
00211111111111111200000
00211111111111111200000
00211111111111112000000
00211111111111120000000
00022222111111200000000
00000000222222000000000
00000000000000000000000
00000000000000000000000
00000000000000000000000
```

3. After Perimeter

APPENDIX C

Flow Diagram



BIBLIOGRAPHY

1. Addressing for the OCR Optical Character Reader, received from Customer Relations Representative, U.S. Postal Department, San Francisco, California

2. Knoll, A.L., "Experiments with 'Characteristic Loci' for Recognition of Handprinted Characters," IEEE Transactions on Computers, v. C-18, p. 366-372, April 1969.

3. Russell, D. H. and Karp, E. E., Reading Aids Through the Grades, Teachers College Press, 1967.

4. Lee, D. M. and Allen, R. V., Learning to Read Through Experience, 2d ed., Meredith, 1963.

5. Harris, A. J., Readings on Reading Instruction, David McKay Co., 1967.

6. Noble, J. K. and Handwriting Research Institute, Better Handwriting for You, California Office of State Printing, 1967.

7. Katsuragi, S., Gerchi, H., Mori, K., and Watanabe, S., "Recognition of Handwritten Numerals Using Decision Graph," Proceedings of the International Joint Conference on Artificial Intelligence, p. 161-170, May 7-9, 1969.

8. Munson, J. H., "Experiments in the Recognition of Handprinted Text: Part I - Character Recognition," AFIPS Conference Proceedings, v. 33, part 2, p. 1125-1138, Fall 1968.

9. Duda, R. O. and Hart, P. E., "Experiments in the Recognition of Handprinted Text: Part II - Context Analysis," AFIPS Conference Proceedings, v. 33, part 2, p. 1139-1150, Fall 1968.

10. Greanias, E. C., Meagher, P. F., Norman, R. J., and Essinger, P., "The Recognition of Handwritten Numerals by Contour Analysis," IBM Journal, v. 7, no. 1, January 1963.

11. Powers, V. M., On-Line Recognition of Hand-drawn Characters from Directional Information, Ph.D. Thesis, University of Michigan, Ann Arbor, 1970.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Asst Professor G. Gibbons, Code 53GP Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
4.	Captain P. L. Townsend MCTSSA Test Bed Camp Pendleton, California	1
5.	Dr. A. L. Knoll Honeywell Data Systems Division-A1319 2600 Ridgeway Parkway Minneapolis, Minnesota 55413	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

A Simplified Approach to Character Recognition

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)
Master's Thesis; June 1971

5. AUTHOR(S) (First name, middle initial, last name)

Patrick L. Townsend

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1971	34	11

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

This thesis describes an off-line character recognition procedure, tested on the ten digits. The basic scheme is an attempt to recognize the character on the basis of a general memory scheme dependent on the overall shape of the entire character. Failing this, a system of checking two specific features, loops and spurs, is called into play on an as-needed basis. This idea of considering specific features only if a more general recognition procedure does not result in a single answer is unique in the character recognition field.

Comparison to other systems indicates that the system is relatively successful, particularly in view of the reduced computer effort expended.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Character Recognition						

DD FORM 1 NOV 65 1473 (BACK)

0101-807-6821

Thesis 128468
T767 Townsend
c.1 A simplified approach
to character recogni-
tion.